VIDEO STEGANOGRAPHY USING LSB TECHNIQUES

Lee Yung Yau Dr Asif Iqbal Hajamydeen

*Faculty of Information Sciences and Engineering, Faculty of Information Sciences and Engineering,*

*Management and Science University Management and Science University*

***Abstract*---- Video steganography is a vital medium for covert communication, which enables the hiding of confidential and secret data within video files while preserving their visual quality. The Least Significant Bit (LSB) technique has been widely used in video steganography due to its simplicity and efficiency. However, existing LSB-based systems suffer from limitations such as low embedding capacity and vulnerability to steganalysis attacks. One of the primary objectives of the project is to increase the embedding capacity further while maintaining the visual integrity of the video. Additionally, with advanced LSB embedding techniques and encryption mechanisms, the systems maximise the amount of data hidden within the video while safeguarding its security and confidentiality. Rigorous testing and evaluation validate the system’s performance and demonstrate its effectiveness in securely embedding and extracting hidden data within video files. The developed video steganography system utilizing the LSB technique offers an efficient and reliable solution for covert communication.**

*Keyword--- Least Significant Bit (LSB), Cover image, Stego-image, Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR)*

1. **INTRODUCTION**

Steganography is the art/science of hiding the existence of the communication between the sender and the receiver. The word steganography comes from the Greek words Steganós (Covered) and Graptos (Writing) and means "hidden writing" [1]. Video steganography is a branch of information hiding focusing on concealing confidential or sensitive data within video frames while maintaining visual quality.

Similar to image steganography, video steganography aims to hide data within the video frames so that the changes are imperceptible to human observers.[2] One widely used technique in video steganography is the Least Significant Bit (LSB) embedding technique. The technique of hidden messages mentioned is used to counter the attacks on data and provide security, privacy, confidentiality, and integrity to the data’s sensitivity. [3]

1. **PROBLEM STATEMENT**
2. ***Payload Limitation***

The Least Significant Bit (LSB) technique for hiding information in video files is constrained by its limited payload capacity. Because the LSB technique alters only the least significant pixel values, it can change only a single bit per pixel, even in an 8-bit channel with 8 bits. This results in a severe limitation on the maximum data payload.

1. ***Inverse Relationship***

There exists an inverse relationship between embedding payload capacity and embedding efficiency in LSB-based steganography. As embedding efficiency increases, embedding payload capacity decreases, and vice versa. This means that attempting to maximize the hidden data capacity can lead to decreased stego-video quality.

1. ***Quantization Effects***

The impact of quantization effects can potentially alter pixel values and result in data loss or degradation, making it crucial to devise innovative approaches to enhance the technique's resilience and protect against potential security breaches.

1. **OBJECTIVES**
2. *User-friendly steganography process by providing an intuitive interface for users to select video files, input and encrypt text, and choose specific frames for data embedding.*
3. *Embed text into video frames using the LSB method but also ensure security through AES encryption, reinforced with user-provided passwords*
4. *Efficiency in handling large video files and maintaining integrity and quality between original video and stego-video.*
5. **SIGNIFICANCE OF THE PROJECT**

The significance of the project lies in its ability to offer a practical solution for secure and covert communication through video files. By leveraging the LSB techniques, this study addresses the need for robust and reliable video steganography techniques, fostering data protection, privacy, and secure communication in variousapplications and domains*.*

Firstly, it enables secure communication by allowing the embedding of secret information within video files, ensuring the confidentiality and integrity of hidden and sensitive data. This is crucial in the modern digital age when privacy violations and unauthorized information access are one of the major concerns.

Secondly, the project addresses the need for covert communication. Covert communication refers to the transmission of information without raising suspicion or attracting attention. By combining video steganography with LSB techniques, users may hide sensitive information within video files, rendering it indistinguishable from the human eye.

This has important uses in several fields, including intelligence gathering, law enforcement, and private commercial communications where covert communication is crucial. The research makes a significant contribution to the development of reliable and effective covert communication systems by successfully implementing the LSB algorithm, guaranteeing that sensitive information may be conveyed safely and covertly without being discovered.

1. *User Scope*

Users of this software tool will be able to efficiently embed and extract hidden information within MP4 video files. The project aims to provide users with the necessary skills and knowledge to use the LSB technique ethically and responsibly. It also demonstrates the real-world applications of video steganography in various industries, including digital investigations, national security, and digital forensics.

1. *System Scope*

The system scope involves the hardware components including video capture devices and basic computer systems, while software components encompass video editing, compression, and decoding software. The system scope also covers the various steps in the steganography process, such as video selection, hidden text determination, frame selection, and LSB embedding.

1. *Assumption(s)*

It’s assumed that the video files used are either uncompressed or compressed using lossless compression algorithms to prevent the loss of hidden information during compression. It also assumes that the video files will not undergo immediate modifications such as resizing or cropping after embedding the hidden data. Additionally, the project assumes the use of the MP4 video format as the default common format.

1. *Limitation(s)*

Mainly the limited capacity to hide data within video files, as it may impact the quality of the video file and vulnerable to steganalysis attacks, potentially compromising the security of hidden information.

1. **LITERATURE REVIEW**

The main focus is on video steganography techniques, with a particular emphasis on the use of the Least Significant Bit (LSB) method. Video steganography plays a crucial role in secure communication and covert data transmission. Several studies have explored the application of the LSB technique in video steganography, each with its advantages and drawbacks. For example, sequential encoding and decoding-based video steganography, achieve high PSNR values but with limited security.[4] One of the different approaches is by embedding two bytes of the message into the first two pixels of each frame, leading to a higher payload but vulnerability to statistical attacks.[5] Studies have demonstrated the trade-offs between imperceptibility, payload, and security in LSB-based video steganography.

Additionally, the literature review covers various aspects of LSB-based steganography, including its simplicity and effectiveness. LSB methods are commonly used due to their ease of implementation. The hiding capacity in LSB methods depends on the number of secret bits (k) that can be manipulated in the cover video, and this capacity can impact both imperceptibility and security.

Moreover, the integration of Reed-Solomon coding has been explored to enhance security and robustness in hidden data, leading to novel approaches in video steganography, watermarking, and data hiding. Each of these methods contributes to the understanding and improvement of steganographic practices.

The literature review highlights the complex landscape of video steganography, particularly focusing on the LSB technique and its applications. It underscores the challenges of balancing imperceptibility, payload, and security in LSB-based methods and discusses the potential of combining LSB with advanced techniques like Reed-Solomon coding to enhance data security and robustness.

*A. Review of Related Literature*

1. *LSB (Least Significant Bit)*

LSB-based methods are commonly used for image, audio, and video steganography due to their simplicity and effectiveness. LSB techniques are described as k-LSB substitution methods, where 'k' represents the number of secret bits that can be hidden. The choice of 'k' influences the hiding capacity, imperceptibility, and security of the steganographic method.[6][7]

*2）AES (Advanced Encryption Standard)*

AES is a symmetric block cipher algorithm that is extensively used for secure data encryption and decryption. It operates on a block cipher principle, where data is divided into blocks and encrypted separately. It supports a range of key lengths, including 128-bit, 192-bit, and 256-bit keys, providing strong protection against brute-force attacks.[8][9]

*3）PSNR (Peak Signal-to-Noise Reduction)*

PSNR is a critical indicator of quality retention in steganographic practices, especially in the context of image and video steganography. Recent advancements have shown significant improvements in embedding capacity and quality preservation by combining the LSB technique with other methods like Huffman chunk coding or pixel value differencing. These enhancements lead to higher PSNR values, indicating better quality preservation in the steganographic process. [10][11]

*4）Reed-Solomon Coding*

Reed-Solomon coding is introduced as an advanced technique to enhance the security and robustness of hidden data in steganography. This coding method is employed to protect data from potential distortions during transmission and compression. Several studies have explored and demonstrated the value of Reed-Solomon coding during the integration of Reed-Solomon coding into steganographic practices, resulting in improved security and resilience. [12][13]

1. **METHODOLOGY**

The development methodology for video steganography with LSB techniques follows an Agile approach, emphasizing iterative and incremental development.

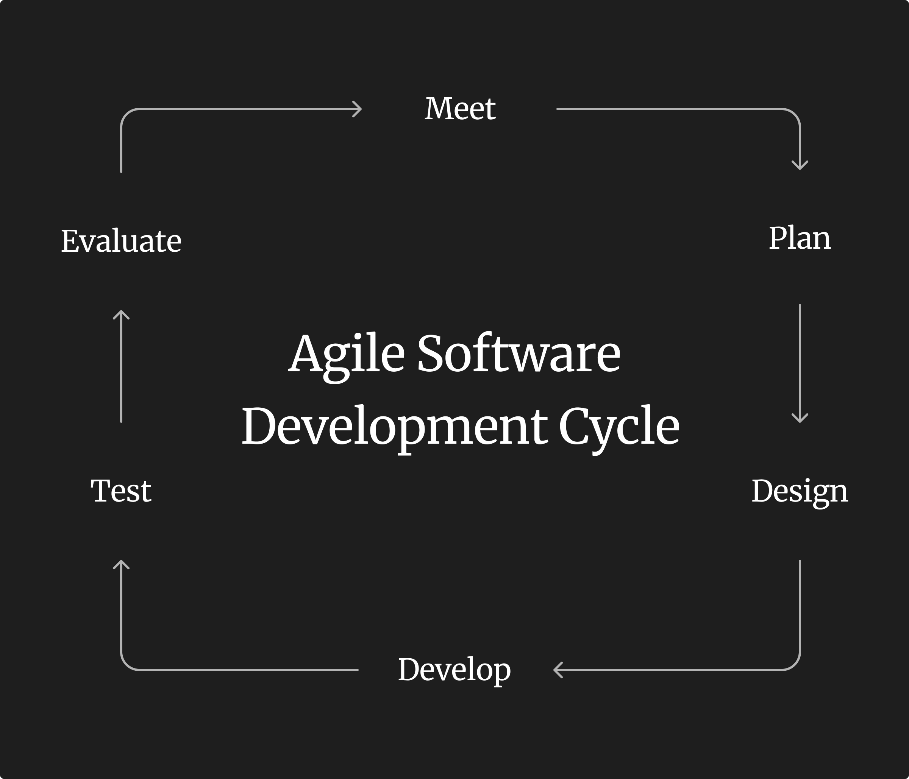


Fig 1: Phases in Agile methodology

*Phase 1: Meet*

Gather the stakeholders involved to discuss the project's goals, challenges specific to video steganography, and the use of LSB for data embedding.

*Phase 2: Plan*

Outline the project scope, define timelines, and organize sprints. Choose the necessary tools and software, and establish the initial approach for implementing the steganography algorithm.

*Phase 3: Design*

Create a detailed design on the steganography system. Specify how LSB will be used to hide data within videos, outline the data encoding and decoding processes, and design the user interface if applicable.

*Phase 4: Develop*

Begin the coding process of the steganography system according to the design plan. Implement and outline the LSB technique.

*Phase 5: Test*

Conduct rigorous testing to evaluate the system's functionality and security. Assess the effectiveness of data hiding and the system's resistance to steganalysis techniques.

*Phase 6: Evaluate*

*Review the test results with the stakeholders involved to assess the system's performance. Discuss any feedback or issues that emerged during testing and make necessary refinements.*

*System Overview*

1. *Graphical User Interface (GUI) Development with PyQt5*

The GUI is built upon PyQt5, which is a set of Python bindings for the Qt application framework. It offers an intuitive interface for users to interact with the application, facilitating tasks like video selection, parameter configuration, and process initiation. It includes input fields for text data and encryption keys, video file selection dialogues, and status messages to guide the user through the process of embedding and extracting data. PyQt5 also provides the means to display progress bars and other notifications to enhance user experience.

1. *Video Processing Library*

Handled by the OpenCV library, which provides a rich set of video manipulation capabilities. It involved reading video frames, modifying pixel values, and writing back the processed frames into a new video file. It allows direct pixel manipulation where the LSBs can be changed according to the binary representation of the data to be hidden. The video is processed frame by frame, and the modified frames are then reassembled into a new video file.

1. *Encryption Techniques*

AES (Advanced Encryption Standard) is employed to encrypt the text before it is embedded to protect against unauthorized access. A user-defined key is used to encrypt the text, which is then encoded with base64 to ensure safe embedding within the video frames. This dual layer ensures that even if the data is extracted, it cannot be deciphered without the correct key, further providing a strong level of security and the confidentiality of the embedded data.

1. *Key Management*

It’s a crucial aspect of the encryption process. The system must ensure the secure creation, storage, and retrieval of encryption keys. The user is prompted to enter a secure key, which is then used in the encryption process. The key is not stored within the application to maintain security but may be required for decryption.

1. *Data Extraction*

This component is responsible for retrieving embedded information from the video. It reverses the process of embedding, and decrypting the extracted data to reveal the original information.

*System Architecture*

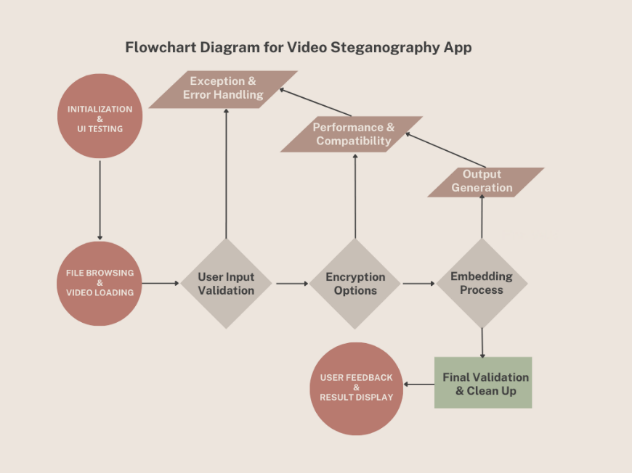


Fig. 2. Halal Food Product Traceability System Architecture

**Fig 2:** Flowchart for Video Steganography App

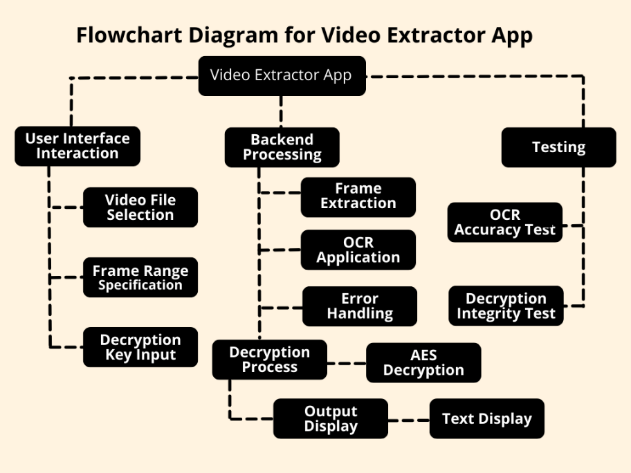
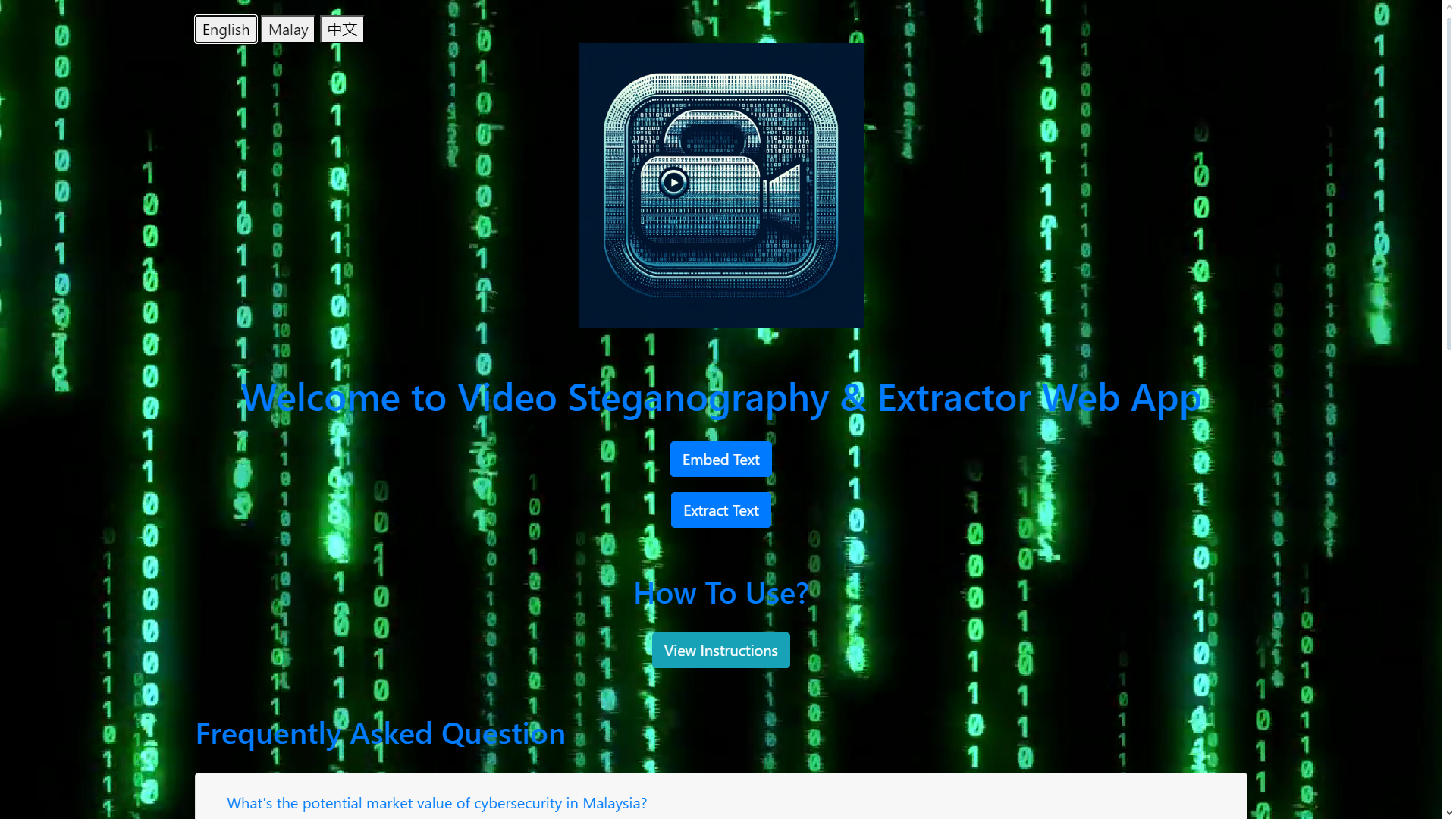
**Fig.2** outlines the process flow for a Video Steganography App, which begins with Initialization and UI Testing, ensuring the app is ready for user interaction. Following this, the user can browse and load a video file, with user input being validated to ensure correct data entry. If errors occur, Exception and Error Handling takes place. Next, the user is presented with Encryption Options to secure the data before the Embedding Process, where the data is concealed within the video. The process then assesses Performance and compatibility to ensure the app works efficiently across different platforms or video formats, leading to Output Generation. The user receives the steganographed video, provides Feedback, and sees the Result Display. Finally, the app undergoes Final Validation & Clean Up to ensure that the output is accurate and all temporary data or processes are terminated properly.

Fig 3 Flowchart for Video Extractor App

Fig.3 illustrates the operational process of a Video Extractor App, which is often for extracting and decrypting text after embedding and encrypting text on the Video Steganography App, the system is divided into three main segments: User Interface Interaction, Backend Processing, and Testing. In the User Interface Interaction phase, the user selects a video file, specifies a range of frames, and inputs a decryption key. The Backend Processing phase begins with extracting frames from the video, applying Optical Character Recognition (OCR) to convert any text within the frames into a machine-readable format, and includes error handling for any issues that arise. The decryption process, specifically using AES (Advanced Encryption Standard) decryption, is then applied presumably to the text extracted. Finally, the Testing phase ensures the app's performance through an OCR Accuracy Test and a Decryption Integrity Test to verify the correctness and integrity of the text extraction and decryption processes. The output, which is the decrypted text, is then displayed to the user.

1. *Video Steganography Web App – Main Page*

****

*Fig.4 Video Steganography Web Main Page*

Fig.4 shows the homepage of the web application with a navigation bar offering language options. The main part of the page highlights the application's function for video steganography and extraction, with options to embed text, extract text, and instructions on how to use the app. It also includes a section for frequently asked questions, with one question visible regarding the market value of cybersecurity in Malaysia.

1. *Video Steganography – User Interface*

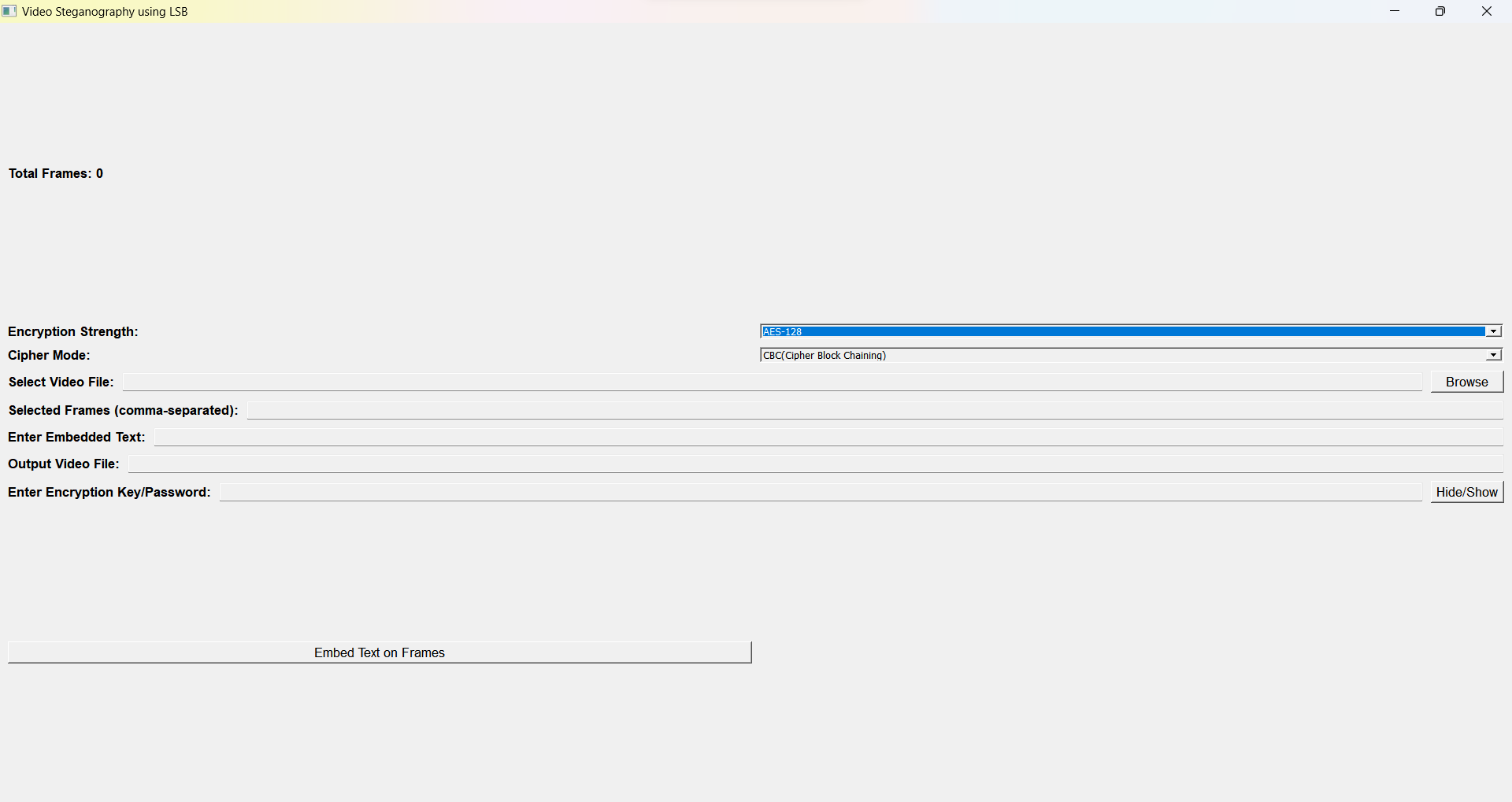
****

Fig.5 Video Steganography User Interface

Fig.5 shows the user interface for embedding text into a video using the Least Significant Bit (LSB) method, which is a common technique in steganography. The interface includes options to select the encryption strength, cipher mode, video file, and the specific frames where the text will be embedded. It also provides fields for the user to enter the text to be embedded and an encryption key or password.

*C Video Extractor – User Interface*

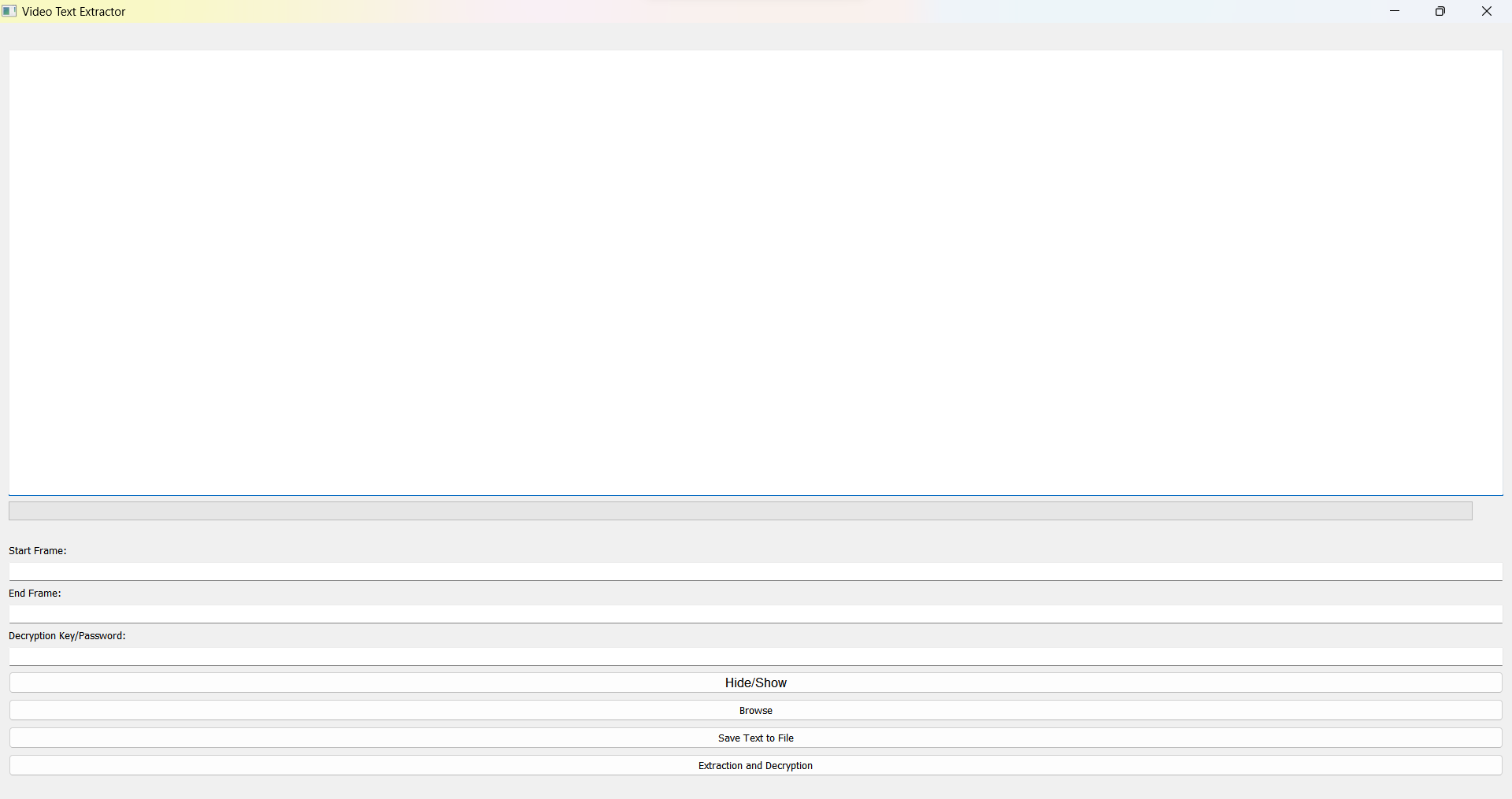
****

Fig.6 Video Extractor – User Interface

Fig. 6 illustrates the interface for the video text extractor part of the application. This interface provides options to select the video file, define the start and end frames for text extraction, and enter the decryption key or password. There are also buttons to browse for files, save the extracted text to a file, and start the extraction and decryption process. This would be used after steganography has been applied to retrieve the embedded text from the video file.

1. **RESULTS & DISCUSSION**
2. *Results*

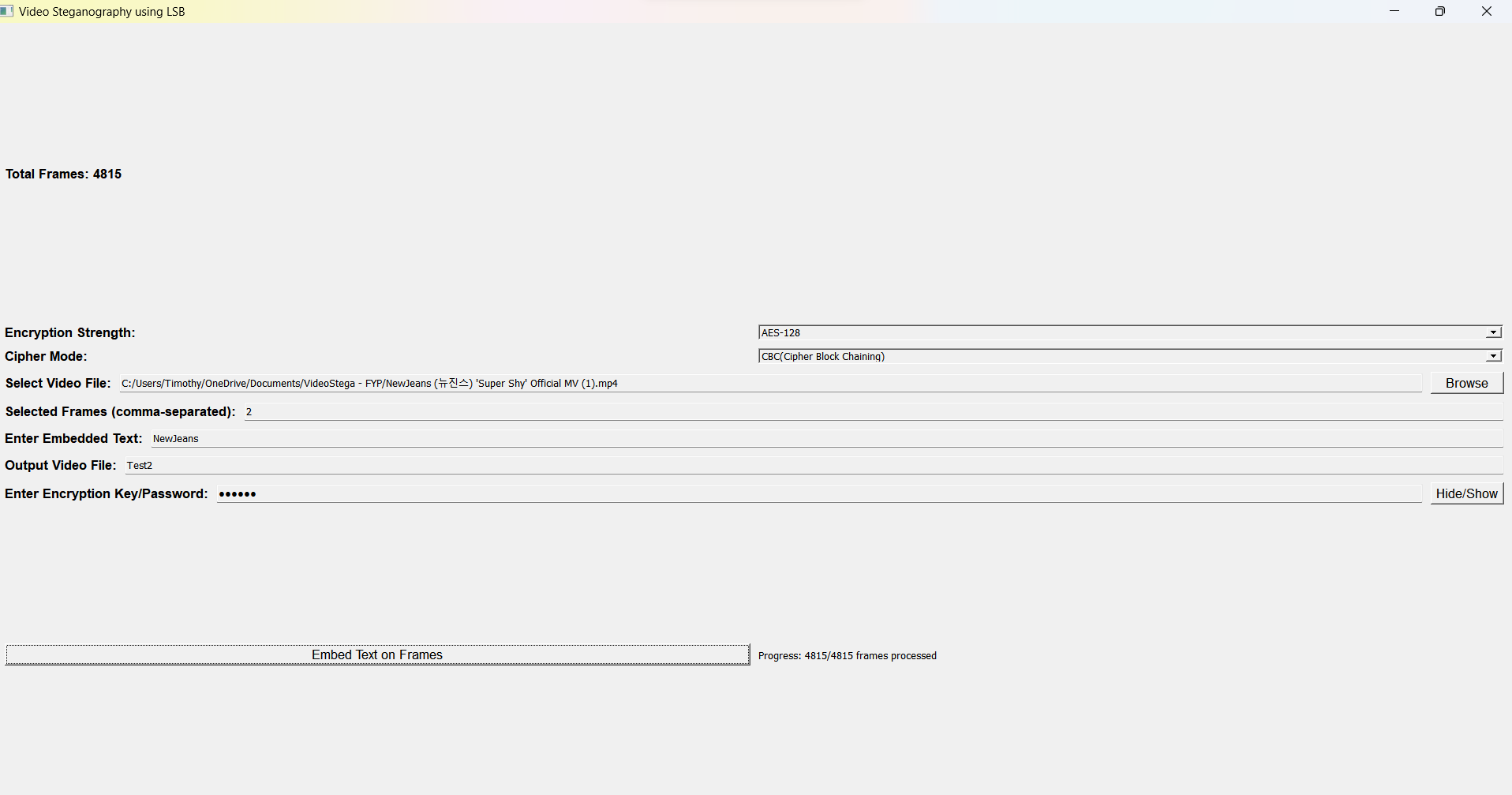


Fig. 7 Video Steganography App with Test Results

In the observed results based on Fig.7, the steganography application demonstrated its capability to effectively process video frames for text embedding using the Least Significant Bit (LSB) technique. The application completed its task by processing all 4816 frames with AES-128 and CBC block. The process log provided by the video processing library validated the successful embedding of the text "test" into the video, culminating in the generation of a new video file with the embedded data intact.

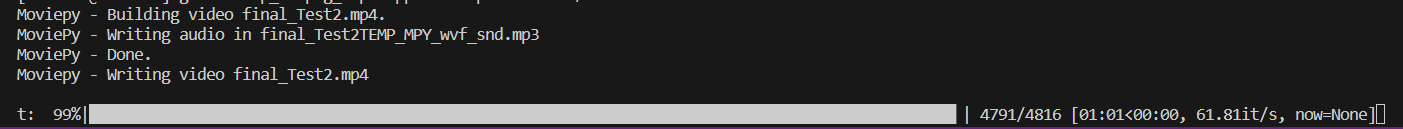


Fig.8 Video Steganography App with Test Results

As from Fig.7, the generated video files were showing successfully embedded hidden text but were observed as no audio playing once played. To achieve a more complete culmination in the generation of a new video file with the embedded data intact with complete audio. MoviePy as a video processing utility and library utilized and processed the video file as shown in Fig.8 for a final video file completely with audio.

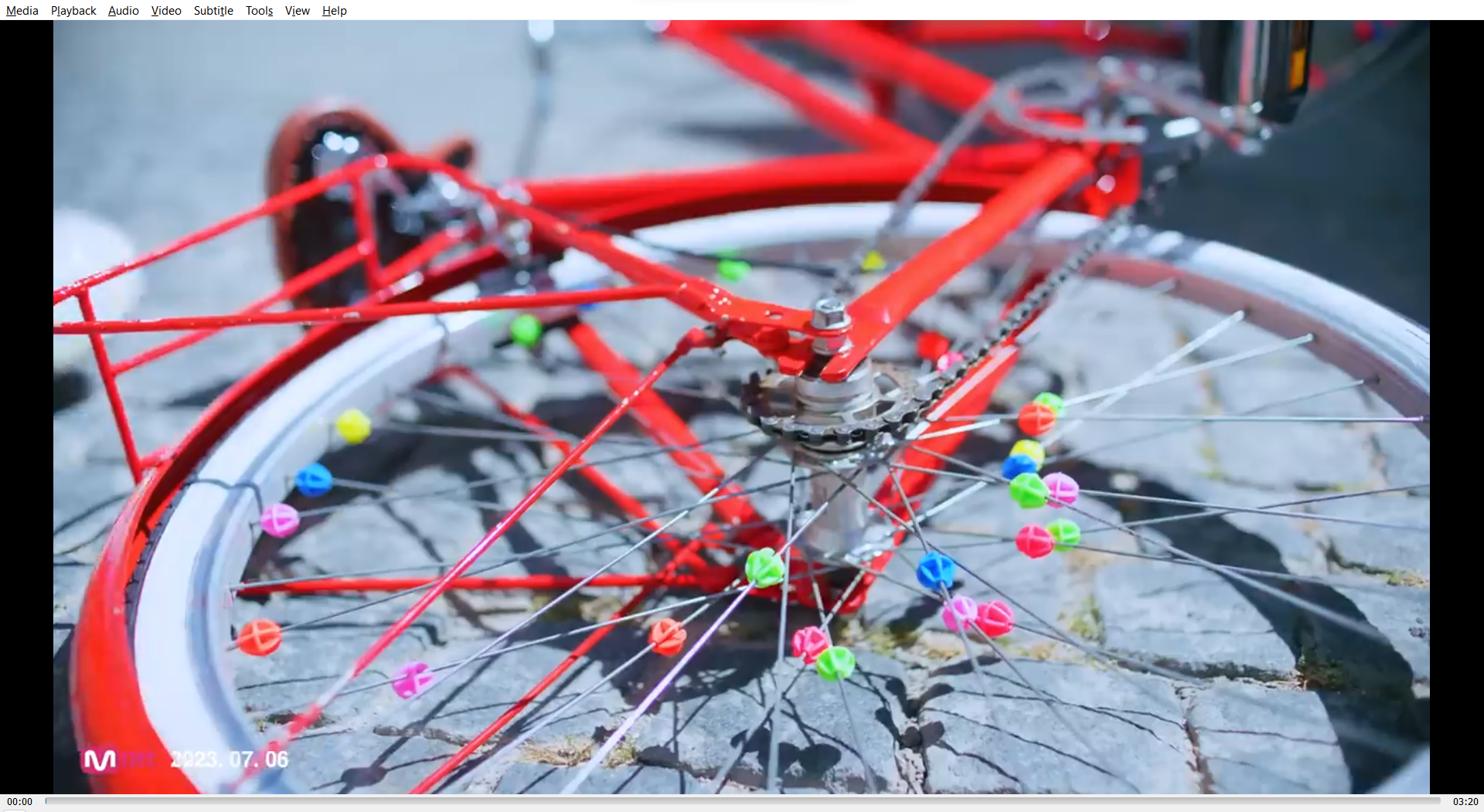


Fig. 9 Original Video



Fig. 10 Stego-video

Fig. 9 and Fig. 10 illustrate the main difference in between the original video and stego- video. The users can extract the text via Video Extractor to select the stego-video file for text extraction and saving the extracted text to a file.

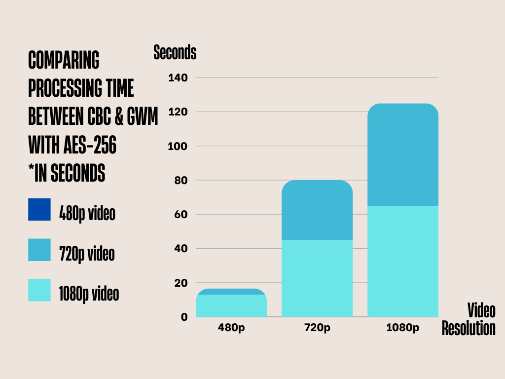


Fig. 11 Processing Time Between Video Resolution

Fig.11presents a comparative analysis of the processing times for videos of different resolutions, encrypted using AES-256. It reveals a clear upward trend in processing duration concerning video resolution, with 1080p videos requiring the most time, followed by 720p, and 480p videos completing the fastest.

**ACKNOWLEDGEMENT**

The authors are deeply appreciative of Management and Science University and its faculty for their invaluable support, which has enabled the dissemination of research findings on Video Steganography using LSB techniques. It is important to note that the authors bear full responsibility for the content presented in this study, and any residual errors are attributed solely to the authors. The viewpoints expressed in this research are those of the authors and do not necessarily mirror the stances of our respective institutions, employers, or funding organizations.

**REFERENCES**

[1] R. Anderson, and F. Petitcolas, "On the limits of steganography", IEEE Journal of Selected Areas in Communications, Vol. 16, 1998.

[2] Kunhoth, J., Subramanian, N., Al-Maadeed, S. *et al.* Video steganography: recent advances and challenges. *Multimed Tools Appl* **82**, 41943–41985 (2023). https://doi.org/10.1007/s11042-023-14844-w

[3] Al-Khater WA, Al-Maadeed S, Ahmed AA et al (2020) Comprehensive review of cybercrime detection techniques. IEEE Access 8:137,293–137,311

[4] Ramalingam, M., & Isa, N.A. (2014). A steganography approach for sequential data encoding and decoding in video images. 2014 International Conference on Computer, Control, Informatics and Its Applications (IC3INA), 120-125.

[5] Kapoor & Mirza (2015) Kapoor V, Mirza A. An enhanced LSB-based video steganographic system for secure and efficient data transmission. *International Journal of Computer Applications.*2015;121(10):38–42. doi: 10.5120/21580-4649.

[6] C. H. Yang, C. Y. Weng, S. J. Wang, and H. M. Sun, "Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems." IEEE Transactions on Information Forensics and Security, 3(3), 488-497, 2008

[7] Kaur, R., Pooja, & Varsha (2016). A hybrid approach for video steganography using edge detection and identical match techniques. 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 867-871.

[8] Daemen, J. & Rijmen, Vincent. (2002). AES the Advanced Encryption Standard. The Design of Rijndael. 26.

[9] Rao, S., Mahto, D., & Khan, D. (2017). A Survey on Advanced Encryption Standard

[10] Than, Shwe. (2020). Secure Data Transmission in Video Format Based on LSB and Huffman Coding. International Journal of Image, Graphics and Signal Processing. 12. 10-17. 10.5815/ijigsp.2020.01.02.

[11] Lee, Haeyoung. (2018). Data Hiding in Spatial Color Images on Smartphones by Adaptive R-G-B LSB Replacement. IEICE Transactions on Information and Systems. E101.D. 2163-2167. 10.1587/transinf.2017EDL8232.

[12] Bhattacharyya, D., Bhaumik, A.K., Choi, M., Kim, Th. (2010). Directed Graph Pattern Synthesis in LSB Technique on Video Steganography. In: Kim, Th., Adeli, H. (eds) Advances in Computer Science and Information Technology. AST ACN 2010 2010. Lecture Notes in Computer Science, vol 6059. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13577-4\_6

[13] Diop, I., Farssi, S., Khouma, O., & Diouf, H. (2012). New Steganographic scheme based of Reed- Solomon codes. International Journal of Distributed and Parallel systems, 3, 81-89. <https://doi.org/10.5121/IJDPS.2012.3207>